AFRL-OSR-VA-TR-2013-0516

ENABLING MORE COMPLEX AND ADAPTIVE SYSTEMS WITH MACHINE AND HUMAN COMPONENTS USING AUTOMATED REASONING METHODS

NICHOLAS CASSIMATIS

RENSSELAER POLYTECHNIC INSTITUTE

09/25/2013
Final Report

AIR FORCE RESEARCH LABORATORY
AF OFFICE OF SCIENTIFIC RESEARCH (AFOSR)/RSL
ARLINGTON, VIRGINIA 22203
AIR FORCE MATERIEL COMMAND

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 23-09-2013	**2. REPORT TYPE** Final Report	**3. DATES COVERED** *(From - To)* July 2010 - July 2013

4. TITLE AND SUBTITLE

ENABLING MORE COMPLEX AND ADAPTIVE SYSTEMS WITH MACHINE AND HUMAN COMPONENTS USING AUTOMATED REASONING METHODS

5a. CONTRACT NUMBER
FA9550-10-1-0389

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Nicholas L. Cassimatis

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
875 N Randolf St, Arlington, VA 22230

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This project aimed to make significant advances towards enabling more adaptive systems involving human and machine components by characterizing system behavior as the result of a reasoning process. Rather than specifying every operation in advance, this approach only requires one to provide the system with its overall goals in addition to some knowledge of the environment, its dynamics and the effects of its actions. In unanticipated or troublesome situations, the system would adapt its behavior by reasoning about the appropriate actions to take to achieve its goals. During this work we helped enable a system that took vague and incomplete commands from a human user and performed the correct action. This was enabled by using a reasoning engine to resolve the ambiguities and infer missing information from the command. The technical challenges was that current inference engines did not scale to problems of this size. We made several advances that enabled them to be used on such problems and demonstrated them on working systems that displayed a notable increase in the abilities of computers to understand natural language commands.

15. SUBJECT TERMS

Adaptive systems, human-machine interface, autonomous systems.

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Nicholas L Cassimatis
a. REPORT public	**b. ABSTRACT** public	**c. THIS PAGE** public	public	1	**19b. TELEPHONE NUMBER** *(include area code)* 518-276-6575

Enabling More Complex and Adaptive Systems with Machine and Human Components Using Automated Reasoning Methods

Final Technical Report

Nicholas L. Cassimatis

1 Abstract

Several of the challenges in creating systems that include both human and machine components involve a mismatch between the characteristics of human cognition and computer systems. Human cognition is able to deal with ambiguity, incomplete information, ill-formed representations, and unexpected changes in the environment. Conventional computer systems, however, must typically have their behavior specified using languages that are fully explicit, unambiguous, and which specify in advance every operation that must be performed.

This project aimed to make significant advances towards enabling more adaptive systems involving human and machine components by characterizing system behavior as the result of a reasoning process. Rather than specifying every operation in advance, the approach only requires one to provide the system with its overall goals in addition to some knowledge of the environment, its dynamics and the effects of its actions. In unanticipated or troublesome situations, the system would adapt its behavior by reasoning about the appropriate actions to take to achieve its goals. To perform this reasoning, we will used Polyscheme framework because of its demonstrated ability to produce cognitive models of human reasoning that have several characteristics uniquely suited to enable adaptive real-time interaction between humans and machines.

This effort will required several research advances. First, elements of the conventional computer languages and representations needed to be formally characterized as parts of a reasoning problem. Second, Polyscheme required three new abilities: In order be able to deal with the side effects of actions, it must be able to reason about causal relations; to handle a changing environment, it must incorporate new reasoning about time; and to handle mismatches between available information and knowledge anticipated in advance, it must be able to perform “implied matching”. These advances will be applied and evaluated in the development of a proof-of-concept system.

2 Introduction

It is a challenge to create systems that involve both human and machine components because of the mismatch between the characteristics of human and machine cognition. Human cognition is able to deal with ambiguity, incomplete information, ill-formed representations, and unexpected changes in the environment. Conventional computer systems, by contrast, must typically have their behavior specified using languages that are fully explicit, unambiguous, intolerant of corrupted structure, and which specify in advance every operation that must be performed.

The following is a seemingly simple example of a commander interacting with the crew of an AC-130 gunship that illustrates a kind of interaction between human and other humans that is a relatively natural, but not yet possible between humans and machines such as unmanned aerial vehicles (UAVs). In this scenario, which actually occurred and was captured by a widely distributed video, commander back at base is directing an AC-130 gunship crew as it searches and deals with Taliban personnel, their vehicles, and the shelters they were congregating in.

The following interaction occurs at time 1:02 of the video:

COMMANDER: “In front of the Mosque, there is three vehicles oriented east-west. Do you see those?”
CREW: “Yes.”
…
CREW: “One of the vehicles is moving right now.”
COMMANDER: You are clear to engage it.
CREW: “Roger”
CREW: “we are clear to eng …”
CREW: “Stand by, do not engage. Monitor”
...

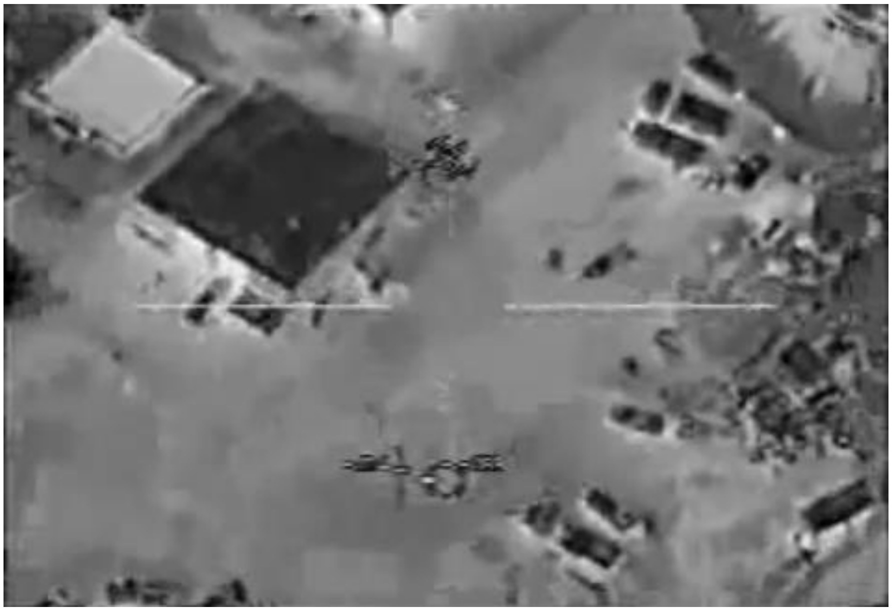

Figure 1. The view from a targeting monitor on an AC-130 during a mission.
Extracted from the video at http://www.youtube.com/watch?v=_OkoWEMCnLQ.

This interaction has several features that enable a level of efficiency that is not currently possible in an interaction between a commander and a UAV. The key difficulty here involves a mismatch between the capability of human cognition and the computer languages used to control nonhuman systems and software. Such languages are specifically designed to be fully explicit in order to determine precisely which operations to execute at a given time, unambiguous, and have very clearly-defined syntax. Further, the above scenario demonstrates that human cognition does not have these properties and that its flaws (from the perspective of computer languages) actually enable quite effective interaction between people in human systems. Specifically:

Embodied and real-time. The interaction above involves the participants having perceptual access to the area they are dealing with. Moreover, they must continually monitor the scene because the world is constantly changing and the perspective of the camera changes as the AC-130 moves. Computer languages and systems often specify sequences of actions in advance and do not naturally alter the behavior of an algorithm they are executing to account for new information from the environment.

Interruptible and retractable. As the crew is confirming the command to engage the vehicle, the commander modifies the order, asserting instead that the vehicle should merely be monitored. Such revisions of plans based on new information are common in human interaction. However, when a computer is issued a command, it typically executes it immediately and there is no opportunity to modify an algorithm as it is being executed.

Incompleteness and real-time responsiveness. When retracting permission to engage, the commander states: "Stand by. Do not engage. Monitor." When saying "monitor", it is implied, but not explicitly stated, that the crew should monitor the vehicle that they were about to engage. In terms of a computer language function call, this is like asserting "monitor" instead of "monitor(vehicle3)", i.e., it is similar to leaving the arguments of a function unstated. In a context where there is time to compile a program in advance and then execute it later, the ability to be incomplete like this is not that important. However, in real-time contexts such as these, if the commander had to say: "Stand by. Do not engage the vehicle that is moving. Monitor the vehicle that is moving", the crew may have performed the incorrect action before the commander had time to issue the full command. Thus, the ability to deal with incomplete command enables much more effective interaction in time-critical situations.

More abstract commands and adaptation. In addition to not explicitly stating which vehicle the crew should monitor, the commander does not specifically state the steps the crew should take to execute this command. In computer language terms, this is the equivalent of not only omitting the arguments from the `monitor()` function, but not specifying the operations that implement that function as well. The ability of human cognition to deal with such omissions has at least two benefits: It leads to much more efficient and real-time communication, and also enables people to be much more adaptive. As the vehicle changes speed and direction as it moves towards and away from the optimal tracking location, the crew can in real-time decide what steps to take to continue monitoring the vehicle. Their tracking of the vehicle would have been much less adaptive and effective if they were simply executing a straightforward computer algorithm unable to adapt to changes.

Ambiguity. In the statement, "you are clear to engage it", "it" can strictly speaking refer to any one of the three vehicles. However, it is clear from the context that the intended referent is the car that is moving. Computer systems are typically incapable of dealing with ambiguity and thus, in this case, one would have to spell out in more lengthy detail which vehicle was intended. Therefore, the ability of human cognition to deal with ambiguity can lead to much more efficient interactions.

Tolerance of corrupted form. The statement that "there is [sic] three vehicles oriented east-west" is ungrammatical ("is" should be "are"). Such minor syntactic issues are often devastating for computer systems (e.g., the difference between "=" and "==" in C), but often –if they are noticed at all– not a problem for human cognition. The ability to deal with "corrupted" form enables people to adapt to imperfect communication channels between them. This ability will be increasingly useful as software systems become embedded and distributed within noisy, mobile or not perfectly reliable networks.

These examples all illustrate that several aspects of human cognition[1] that superficially appear as flaws compared to computer language and architectures actually have several benefits that enable more efficient, adaptive, and complex real-time interaction. It is natural therefore to consider using methods in artificial intelligence and cognitive modeling to implement computer systems, since one of the goals of each of these fields is to endow computers with human-level intelligence.

While the various methods used in this field each make progress towards our goals, they also continue to have several shortcomings that require further research. For example, many cognitive architectures are based on production rule systems. While these do enable some flexibility over traditional computer languages, they still require the exact action to take to be specified for any given scenario. However, in practice it is extremely difficult to predict all of the relevant rules for a complex task in many dynamic domains. Such problems also exist in many artificial intelligence areas. For example, the problem just described corresponds to the qualification problem (McCarthy, 1980) in artificial intelligence.

Planning algorithms (e.g., (Hart, Nilsson, & Raphael, 1968)) are another approach in artificial intelligence for gaining some flexibility. These allow one to specify the actions (along with their effects) possible in a domain and can in many cases find the appropriate sequence of actions to take to achieve a goal. For each specific kind of situation, one does not need to specify which action to take. Instead these algorithms determine the right actions on their own. While such algorithms represent significant progress towards our goals, they still have some shortcomings. For various reasons of computational complexity (discussed below), planning systems often require all objects in a domain and all the effects of every action to be known in advance. They often require perfect knowledge of the initial state of the environment. They have considerable difficult dealing with situations where the environment is changing. Of course, each of these conditions are quite common in many of the application domains in which we would like to apply modern systems.

The field of behavior-based robotics or embodied systems (e.g., (Brooks, 1991)) was a reaction to these problems. At bottom, however, these were mostly equivalent to (nested) rule-based systems with the main innovation being that the rules were triggered by actual sensory events and actions, not simply by symbolic data structures. While this innovation did enable some new applications, success has mostly been limited to basic

[1] It should be noted that while it is often simpler to illustrate these issues with cases involving communication, human cognition and human interaction are impressively complex and adaptive in most all of its manifestations, whether these involve language or not.

sensorimotor applications (such as crawling ant-robots and the Roomba). The limitations of purely rule-based systems have prevented these approaches from having been useable in more complex and demanding applications.

To summarize, many of the difficulties involved in organizing systems with human and machine components stem from the mismatch in the abilities of human cognition and those of languages and data structures for computer systems and software. While existing artificial intelligence and cognitive modeling methods have made some progress towards resolving this mismatch, there is still considerable work to do.

3 Complex, adaptive real-time systems through reasoning

The properties of human cognition that we have been discussing are enabled in part by their reasoning abilities. We thus propose that by embedding these abilities within computer systems, we may enable significant advances in how humans and machines can interact within larger systems. In this section we first describe how to reformulate a (human or nonhuman) agent's interaction as solving a reasoning problem. We then show how this suggests solutions to many of the problems motivating this proposal to be addressed.

Before proceeding, it will help to clarify what we mean by solving a reasoning (or inference[2]) problem. We can conceive of an agent as having some background knowledge, some goals, and some perceptual information about the environment. The task of an agent is to find a set of actions such that given background knowledge and perceptual information, the goals are solved. That is, it wants to find *actions_taken* such that the following inference can be made:

$$\mathit{background\_knowledge} \wedge \mathit{perceptual\_information} \wedge \mathit{actions\_taken} \rightarrow \mathit{goals\_achieved}$$

As a simple example, consider the task of a UAV tracking a vehicle. The background knowledge (of the operator) includes the characteristics of the vehicle (such as its range of speed and the terrain it can traverse), the perceptual information includes the camera input and various instruments within the UAV, and the actions it takes involve flight commands to the UAV and manipulations of the instruments. If in a particular situation one can infer from the background knowledge, perceptual information, and actions taken by the UAV that the vehicle will continue to be tracked, then the UAV is operating successfully.

While superficially this formulation resembles classical planning and multi-agent system frameworks (Wooldridge, 2002), there are many differences, which were mentioned above. These include the need for complete information of world state, a static environment, and complete knowledge of the effects of actions. Thus, while in a very broad sense we are describing a planning problem and existing planning approaches have much to be learned from, we cannot rely on them exclusively. More research is needed, as will be described below.

We now illustrate using the example from the last section how formulating the agent's role within a larger system as a reasoning problem can address some of the problems we have been describing. For now, we assume that we have a reasoning

[2] While the terms "inference" and "reasoning" are often used in different literatures, we will use the terms interchangeably.

mechanism, which we will call *an inference engine*, that can solve the problems we have described above. No such inference engine exists at present; developing one is an aim of the proposed research.

Ambiguity. Recall that when the commander above said, “you are clear to engage it”, there were three possible vehicles he could have meant by “it”. However, just before, the commander had mentioned a particular vehicle that was moving. One can infer that he wouldn’t have pointed that one out specifically if he hadn’t intended to take some action on it. Further, because there is more urgency in engaging moving vehicles before they escape –this is part of background knowledge– the commander most likely would have given the command to engage the one moving vehicle, rather than the stationary vehicles. Thus, ambiguity (in this case about goals) can be resolved by reasoning about perceptual information and background knowledge.

Incompleteness. When the commander said “Stand by. Do not engage. Monitor.”, he did not specify what should be monitored. However, one can infer by reasoning about the background knowledge that if the vehicle was interesting enough to be potentially engaged, then it would be interesting enough to monitor.

More abstract commands. The commander can simply indicate that the vehicle should be monitored without specifying every step of the monitoring process. This is a straightforward consequence of applying the reasoning abilities of the AC-130 crew; their background knowledge includes information about the capabilities of the AC-130 and training on how it will behave in certain kinds of circumstances. Using this knowledge, they can infer which actions will keep the tracked vehicle in range.

An additional benefit of designing systems using reasoning methods is that that in many cases they provide correctness guarantees. For many systems, one can be confident that the inferences they produce are “sound”, i.e., that they make no incorrect inferences. Some systems also provide “completeness”, i.e., the guarantee that they will make *all* the correct inferences. These guarantees remove some of the brittleness often associated with software. When the knowledge a system relies on changes, sound reasoning algorithms will still provide correct inferences.

4 Implied matching problem

The technical challenges was that current inference engines did not scale to problems of this size. We made several advances that enabled them to be used on such problems and demonstrated them on working systems that displayed a notable increase in the abilities of computers to understand natural language commands. These advances were enabled by dealing with the implied matching problem.

4.1 Definition

This mismatch between the information which is presented to an agent, and the information which the agent expects will be called “The Implied Matching Problem”. In particular, the information that is present in the environment might be implicitly equivalent to the information the agent is expecting, and yet because it exists in an unexpected or unplanned for format, exact matching is not possible. While this difficulty might seem simply a detail to resolve when specifying the system, such pre specification is not exhaustively possible the in real world. The reason for this is clearly elucidated by Keith Devlin:

> Information, as we usually encounter it, is not unlike a ‘bottomless pit’, seemingly capable of further and further penetration. To borrow a term from another fairly new area of mathematics, we might say that the information has what appears to be a *fractal* nature. On the other hand, cognitive agents deal (at any one moment) with a relatively small collection of specific *items* of information extracted from that fractal-like environment. The acquisition of information from the environment by a cognitive agent is a process analogous to, though not necessarily the same as, going from the infinite and continuous to the finite and discrete. (Devlin, 1991, p16).

In fact, the very mechanisms of perception and cognition can be construed as the process of the environment being made available to the agent, and the agent extracting a small quantity of useful information from the almost infinite array of possibilities available to it. Associating this process with analog to digital conversions in electronics, Devlin states “A *cognitive* agent is an agent that has the capacity of *cognition* in this sense; i.e. the ability to make the analog to digital conversion” (Devlin, 1991, p17-18).

If an agent has not been built to make this conversion exactly in the way it is expecting it will encounter a discrepancy between what it experiences and what it expects. Robustness demands flexibility, and this flexibility must be situated in the deepest parts of the agent’s capability to reason.

This problem of “unexpected inputs” exists not only in the external world of concrete objects but extends to the inner world of abstract thought. An example of this phenomenon can be shown by considering counterfactual reasoning. Many verbal statements refer to a counterfactual situations, for example, saying “If I was rich, I would not drive an old car” when it is clear that I am not rich, and my car is indeed old. The difficulties in accounting for how this reasoning works in a formal and consistent way have inspired a great deal of philosophical and linguistic research. Classic attempts include metalinguistic approaches (Goodman 1947), possible world semantics (Lewis, 1973) and mental spaces (Fauconnier, 1985). Most of these frameworks imply a separate “space” for abstract/imaginary reasoning, which is connected to but distinct from the “real” space of perceived reality. These spaces will be termed “worlds” for the purposes of this paper. When attempting to do inference in a computational system over counterfactual statements we have found it useful to describe a distinct counterfactual world, similar to and yet distinct from the real world. In a counterfactual world, the properties of the “real” world are assumed to apply except for a few contrary to fact assumptions and their implications. To avoid creating logical contradictions, each piece of information must be explicitly associated with a world (whether real or counterfactual), and inference rules must take into account the world when matching is done over their terms. In other words, reasoning about the real world should not take into account counterfactual arguments, and reasoning about the counterfactual world should not take into account real world arguments. This however is extremely inefficient because, in general, counterfactual terms will make up only a small quantity of the information in a world. (The world where I am rich presumably still shares a great deal in common with the world where I am not). In an ideal situation, the counterfactual information would simply contain the exceptions, and the rest of the information could be inherited by default from the real world. However, as described above, preventing contradictions requires preserving the information distinct to each domain, and thus inference rules require that all of the items which match a given rule must come from the same world. Thus, unless all of the non-counterfactual information is transformed from the real world to the hypothetical world, the inference rules will not be able to match. This is essentially the same problem that was observed with perceptual information.

The need to reason over information which is distributed between multiple domains is not limited to counterfactual reasoning. Bringing together separate conceptual domains is the fundamental operation involved in conceptual blending (Fauconnier & Turner, 1992), which has been proposed as a fundamental human cognitive capacity enabling creativity, language, and problem solving. "Although language has been said to make an infinite number of forms available, it is a lesser infinity than the infinity of situations offered by the very rich physical mental world that we live in….The extraordinary evolutionary advantage of language lies in its amazing ability to be put to use in any situation" (p178-9). With regard to counterfactuals in particular Fauconnier and Turner conclude "there is no form of causal inference in the social sciences that does not depend upon counterfactual reasoning" (p218). In a similar vein, research on metaphors (Lakoff & Johnson, 1983; Lakoff & Johnson, 1999) has showed that they are ubiquitous in human cognition, and that they operate by extending well known concrete domains from direct human experience to more abstract ones. Linguistic and semantic studies (Talmy, 2000) have shown that the structure of language and thought is dependent on a framework of basic spatial and force-dynamic interactions which act to organize other conceptual domains. Theories of conceptual reasoning based on simulation (Barsalou, 2009) also assume that information about a situation from memory can be retrieved and used in real time for pattern completion and prediction. Given that no situation is ever exactly the same twice, this assumes that information which was relevant in the past can be matched with new information coming in through perception. In all of these examples, a particular framework with rules and defining examples (which can be thought of as constituting a "world") has to be extended to apply to foreign and novel situations which may be appropriate in some ways, but might also have unique behaviors which need to be overridden. Successfully reasoning over "mixed" domains is thus a critical human cognitive ability which takes place effectively and efficiently.

Building complex systems is difficult, and the human brain is arguably the most complex object yet to be encountered in the universe. The search to understand it by replicating its functionality should therefore be expected to be enormously difficult. At any given stage in development there are numerous challenges which must be solved. Even before complex systems are fully operational, benchmarks can be set to demonstrate incremental progress on particularly challenging aspects of problems. However, care must be taken that the progress is made on fundamental problems, so that success in the project will constitute a true advance in the field (Cassimatis, 2006). We believe that the above examples demonstrate that the Implied Matching Problem is just such a problem, and solutions to the problem, even in limited domains, will represent a major advance in artificial intelligence, facilitating improvements in reasoning over both concrete and abstract situations.

4.2 Alternative Solutions

We must first consider a few objections to the approach by discussing some potential alternative solutions to the issues described above. First, if all of the information needed to match is simply implicit in the observed information, then why not simply transform the observed information on demand and do the match directly? It is true that this is not a purely theoretical problem, as all input encountered in the world could be transformed into all possible expected formats, and matching could proceed. However, in terms of building intelligent systems that can operate under real time constraints, this is not only impractical, but also intractable. First consider the case where all inputs are automatically converted into all possible matching terms. For any given object that is observed, that object would need to be pre-classified for every situation that it might exist in, from the trivial to the arcane. Even when limiting ourselves to a set of relations such as categorical classification, the list of possible roles a given object might play is enormous. In this way, even a small set of information would quickly become an unmanageable data set.

Another solution might be to restrict the possible inputs to a predefined set or to place careful limits on the allowed transformations of given objects. However, human agents do not seem to have this restriction, as their powers of improvisation and creative problem solving attest. This approach also makes intelligent systems vulnerable to the charges of "hand crafting" and existing in "toy worlds".

A final idea on how to get around this difficulty is to detect the situations where there is a mismatch between the expected and encountered information, and do selective transformation in those cases. This idea, while promising, has several downfalls. First, with a large rule set, simply knowing the "needed" transformations for every piece of input coming into the system is a non-trivial task, (potentially larger than the exhaustive list of categories themselves). Secondly, the very enterprise of transformation is built on certainty. If a given categorization were only probable, then the transformation would need to be done contingently on the possibility being true, so that if it were proved false the categorization could be removed. Performing this operation regularly would require significant bookkeeping about which information led to which transformations. In summary, transformation of the incoming information leads to intractable complexity, while restrictions on the incoming information reduce the flexibility and power of the system.

5 Approach

5.1 The Cognitive Substrate Hypothesis

Before describing the proposed solution to this problem, it is necessary to justify our focus, namely on a small set of core relations where the Implied Matching Problem appears. The specific question is: given the vast number of specialties a computer might have (Jeopardy question answering, chess playing, integral solving, etc…) why select any particular subset of relations for emphasis over others. What makes these relations special, and what other relations might also need to be explored? The problem is that human intelligence existed long before chess, Jeopardy, and mathematics were invented. It is unlikely that the biological structures which humans use to perform cognition in these domains existed to solve those problems (even though they tend to be used as "exemplar domains" for brilliant humans to show off). This analysis is built upon the Cognitive Substrate Hypothesis.

> The hypothesis states that there is a relatively small set of computational problems such that once the problems of artificial intelligence are solved for these, that is to say, once a machine, called here a "cognitive substrate," is created that effectively solves these problems, then the rest of human-level intelligence can be achieved by the relatively simpler problem of adapting the cognitive substrate to solve other problems… Progress on the comparatively small (but not trivial) set of problems required to implement a cognitive substrate would constitute progress toward human-level intelligence in all domains (Cassimatis, 2006, p46-47).

Once a common set of substrate functionality is established, other more abstract forms of reasoning can be mapped onto the substrate relations. Research is ongoing to identify potential substrate domains, but the current best guess includes relations such as categorical information, temporal reasoning, and simulation of counterfactual worlds (p48-49). Cassimatis has demonstrated the power of the substrate hypothesis, by showing how a computational framework built around physical reasoning relations can be used to do sentence parsing by a building a mapping between the domains (p53-54). A representation of this is shown in Figure 1.

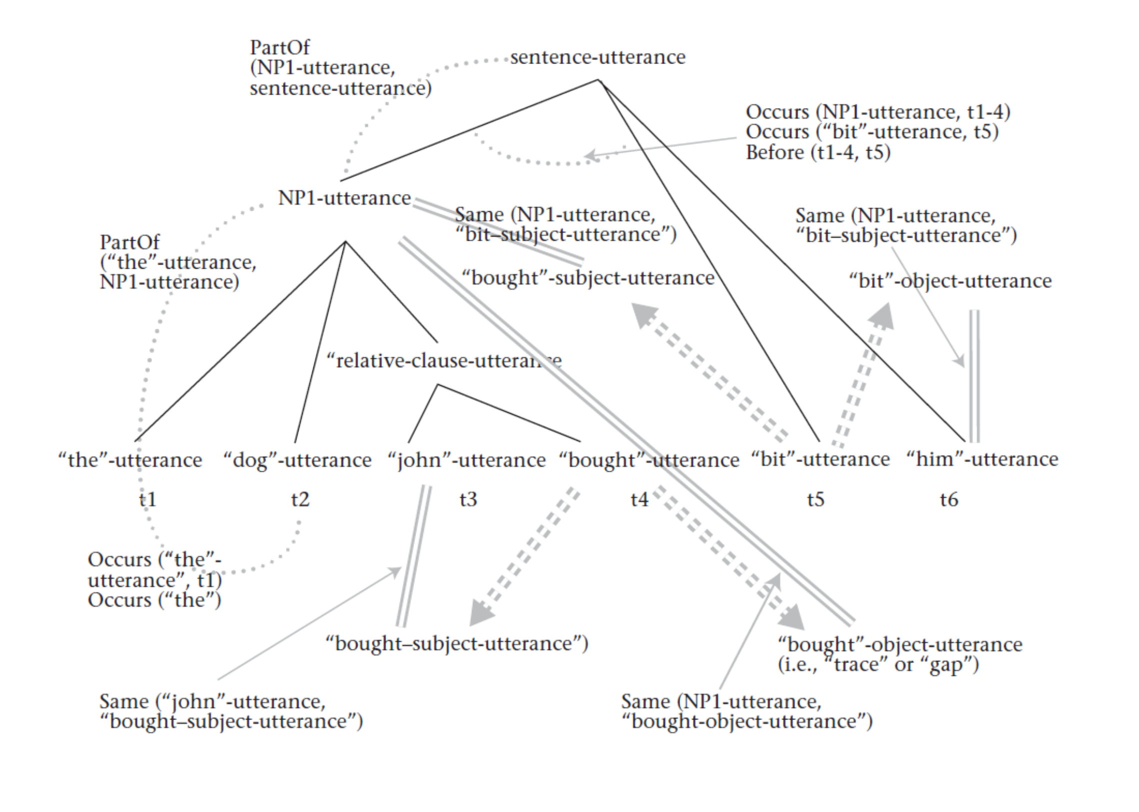

Figure 1: The Syntactic Structure of a Sentence Represented Using Concepts from Infant Physical Reasoning (Cassimatis, 2006)

If the Cognitive Substrate Hypothesis is correct, then it explains why the Implied Matching Problem emerges so frequently among substrate relationships, and it justifies a solution to the Implied Matching Problem which is focused on these domains. For example, implied matching over categories could be used in any domain which is built upon a rich hierarchy of descriptive sets. Implied matching between worlds would facilitate the comprehension of hypothetical verbal statements and any scenario where two separate conceptual domains must interact by overriding one or more of the relations within those domains which conflict.

5.2 The Polyscheme Cognitive Architecture

The cognitive architecture which takes into account the Cognitive Substrate Hypothesis is Polyscheme (Cassimatis et. al, 2010). Polyscheme is a hybrid architecture, integrating multiple types of computational algorithms, in order to leverage both generality and efficiency (p.1). Polyscheme consists of a series of modules termed "specialists" which contain their own algorithms and data structures, but are forced to act together via a shared focus on attention, and the implementation of common functions (pp. 4-5). One of the problems with implementing a cognitive substrate is that reasoning in different domains typically requires extremely different computational techniques (Cassimatis, 2006, p.5). The integrated hybrid nature of Polyscheme enables close interaction between the modules, while allowing each to maintain its own specialized internal structure. Thus, Polyscheme is an ideal platform for demonstrating and exploring solutions to the Implied Matching Problem. A solution for implied matching in one domain, such as categorization, can be implemented as logic unique to a subset of specialists, and then be used to support inference across any conceivable domain.

5.3 Notation

The notation used for examples in the rest of this thesis is intended to be general, but is similar to that used in reasoning in Polyscheme. The notation will describe the relations that hold between objects, their properties and the relations that hold between them. An atom expresses a relation over one or more entities[3] and takes a truth value in world.[4]

1) Predicate(a_1, …, a_n, world)

An atom written by itself asserts that the atom is true; the negation operator (-) indicates an atom is false[5].

"John is holding the ball"

2) Holds(john, ball, w)

"Meg is not holding the block"

3) -Holds(meg, block, w)

Constraints are used to express dependencies between atoms, using operators for conjunction (^), and implication (→). Entities (including predicates) can be given universal force by designating them as variables, by prepending a "?".

"All large dogs bark."

4) IsA(?dog, Dog, w) ^ Large(?dog, w)→Bark(?dog, w)

[3] In this framework, the definition of entities is very general and can include objects, states, sets, and relations

[4] Unless specifically noted otherwise truth values hold in R.

[5] In order to increase clarity, this account assumes Boolean truth values. Nothing, however, in the following analysis precludes the use of more complex truth values or probabilities.

The above constraint is a *hard constraint*: when all the atoms the match its antecedent are true, the implied consequent must also be true. Some constraints can be broken, but at a cost. Such *soft constraints* are indicated by prepending the implication symbol with a cost for breaking the constraint. The sum of the costs of the broken constraints in a world produces the overall cost for the world.

"All large dogs usually bark."

5) IsA(?dog, Dog, w) ^ Large(?dog, w)(.75)→Bark(?dog, w)

While a variable in a constraint that appears in the antecedent has universal force, a variable that appears in the consequent of a constraint has existential force.

"For every large dog there exists a collar that it wears."

6) IsA(?dog, Dog, w) ^ Size(?dog, large, w)
→Wears(?dog, ?c, w) ^ IsA(?c, Collar, w)

This notation straightforwardly maps onto logic. Example (6) can be represented with quantifiers as follows:

"For every large dog there exists a collar that it wears."

7) $\forall x \exists y (Dog(x) \wedge Large(x) \rightarrow Collar(y) \wedge Wears(x, y))$

This notation was chosen for representational parsimony and compatibility with existing inference systems such as production rule systems, and constraint satisfaction algorithms. This is neither a claim that human reasoning uses logic, nor a claim that logic is the best way to implement a world-based framework. For example, previous work has shown (Cassimatis, et al 2004; Cassimatis, et al 2010) that world-based reasoning can be utilized in a system that includes non-logical inference mechanisms. It is also clear that regardless of the system used for representation, an agent must have states that in some way “represent” various states of affairs both internal and external (such as the way things are, and the way things might be), and that these states must be distinguishable from one another to permit the agent to perform appropriate behaviors.

6 Implied Matching Through Conversion

6.1 Implied Matching over Perceptions

A solution to the Implied Matching Problem was implemented in Polyscheme as an addition to the system which performs pattern matching and implication . Polyscheme allows for integration of many types of inference, including a system which implements first order logical inference. For example, "IF someone is a man, THEN they are mortal" could be notated as follows in Polyscheme:

8) IsA(?x, Man, R[6]) → IsA(?x, Mortal, R)

In this rule, membership in the class of "Man" is an antecedent. Membership in the class of "Mortal" is a consequent. Only situations which *exactly* match the antecedents will trigger the inference of the consequent. It is the general inflexibility of this type of matching behavior which causes the Implied Matching Problem. When solving the Implied Matching Problem over perceived inputs, it is important to note that the problem arises specifically because of the requirement that a single object must have all of the required antecedent properties. There can be no exceptions. Some of these properties are obvious, but implied. If someone is a member of the category "Bachelor" they are by implication a "Man" as a bachelor is simply a subset of the category of Men which comprises all of the members who are not married. Therefore, to make the rule above "flexible", we need to modify it so that subcategories will match parent categories. Rewritten, the rule can state something along the lines of "If an object exists in any of the subcategories of "Man", then that object is mortal". This transformation requires the addition of "trivial" Subcategory relations, so that each category is listed as a subcategory of itself (otherwise, the match would fail in exactly the cases where it previous succeeded), and the transitive closure of all possible subcategories in the hierarchy. This is advantageous since a category hierarchy is generally very static and not re-defined during real time inference.

The advantageous effect is twofold – first, the flexibility has been moved to the matching phase, and not the perception phase. Given the difference in quantity between objects that might be perceived, and expectations that the agent has, this is a dramatic improvement in efficiency. Secondly, the representational transformation is moved "off line" to facilitate dynamic real time processing. The agent's expectations are far more static than the situations it encounters, and the transformation can be done before the encounter[7].

Several modifications to Polyscheme were required to facilitate this behavior. The matching system was updated to include the ability to match over subcategories for any situations where categorization was needed for inference. New categorization information

[6] R represents the "Real World"

[7] An obvious disadvantage is that the agent will need to re-integrate new information and changed expectations which could be potentially be costly in a dynamic situation, leading to impaired performance. Anecdotally, humans do indeed seem to have this difficulty when encountering new information and changes of expectations "on the f ly".

was also exhaustively added to the system across their transitive closure to allow for any possible ontological connection. Finally, to enable matching in the trivial case (where the perceptions match the expectation) the trivial sub-categorization (previously ignored) were added to the system.

6.2 Implied Matching over Worlds

World matching presents a different technical challenge than perceptual matching. A full treatment and justification of the use of “worlds” in reasoning can be found in Scally, Cassimatis, Uchida (2012) – the present report seeks to describe a possible implementation of world reasoning.

To explain how this behavior is facilitated by constraint transformation, it is first necessary to describe in detail what a hierarchy of worlds looks like. In the Polyscheme Architecture, the base level world in any given model is “R”, which corresponds to the “real” world, namely the set of information which can be directly observed. From the real world, it is possible to create worlds which inherit all of the information from “R”, and add a specific set of “counterfactual basis” elements, which define the properties of a counterfactual world in contradiction (or in addition to) those which hold in R. For example, in the real world, if I drive an old Honda Civic, I can explore the counterfactual world where I drive an Audi. In this counterfactual case, I want to be able to use all of the information from the real world which has not been overridden. For instance, my place of employment will (probably) not change because of the vehicle that I drive, and neither will the fact that I am a licensed driver. However, my driving style and car washing habits might change drastically. Hence, there is a situation where information from the real world needs to be inherited “by default”, and yet can still be overridden if information that is inferred in the counterfactual world contradicts it.

As a concrete example of how this plays out, take a simple constraint: “If someone owns a sports car, they will drive fast”. This can be expressed symbolically as

9) Owns(?x, ?y,?w)^ IsA(?y,SportsCar,?w)→Drives(?x, Fast,?w)

If it is true that John owns his car in the real world, and that in his dreams John’s car is a sports car, then in his dreams, John will drive fast. At least that is the assumption an agent should make. The problem with implementing this behavior in an artificial system is matching. Namely, we need the proposition for John owning a car to be explicitly stated as being true in his dream world. Otherwise, the match does not happen.

There are two additional requirements for constraint conversion over worlds. First, when we define the counterfactual world, it must be placed in a hierarchy of relevance, for which inheritance is allowed in a single direction. So, in this case, we can state that the dream world of John is “relevant to” the real world. Information about the real world then can be used to reason about John’s dreams. However, by default, information in John’s dreams will not be used to reason about the real world. (This behavior is possible in certain situations, but it cannot and should not be accomplished through a mechanism of default inheritance.) This world tree is not restricted to a single level, though things get progressively more complex as the hierarchy is extended. This might involve the dreams which John has within his dreams, or my beliefs about your beliefs about my beliefs.

Secondly, we need the ability to restrict which information can be propagated from the real world to a counterfactual world. This is done by defining an "override" term, which states whether a piece of information in a given world can be treated as applicable for a counterfactual world higher up the hierarchy. In this case, if John owned a sports car in real life, and has a nightmare where he owns a rusty pickup, then the piece of information IsA(carOfJohn, SportsCar,R) would not be applicable in the world "johnsDreams". This is represented in Polyscheme by the term:

10) IsA(_OVR_, carOfJohn,SportsCar, R, johnsDreams)[8]

Directly interpreted, this means that the proposition IsA(carOrJohn,SportsCar,R) is overridden in the world "johnsDreams". This information is controlled in Polyscheme by a specialist called "Override Specialist" whose responsibility it is to detect and to issue opinions on the truth or falsify of any override terms.

With this framework in place, it is possible to do the conversion into something that states "If someone owns something in a world which is relevant to the world which we are considering, and if that statement has not been overridden, and if that something is a sports car in a world which is relevant to the world which we are considering, and if that statement has not been overridden, then that someone drives fast in the world which we are considering."

When "implied matching support" is turned on in Polyscheme, the conversions described above are done automatically for every constraint that is loaded into the system. The benefits described above, such as representational flexibility, and moving "on line" work to "off line" transformation, were used to run models of natural language processing, beliefs, and blending between counterfactual domains. It has significantly simplified existing models, reduced overhead for better scaling, and has given Polyscheme new reasoning abilities. Polyscheme's capabilities for implied matching, when combined with its ability to reason probabilistically, use quantified objects, and create new objects "on the fly", make it unique in the space of cognitive architectures.

Figure 7 visually demonstrates the difference in complexity of a world reasoning model with and without Implied Matching support. The model on the left was developed for Polyscheme by Paul Bello to demonstrate reasoning the false belief task. All of the operations to transfer beliefs between worlds were written explicitly. The model on the right is a version of the false belief task that uses Implied World matching.

[8] This representation in unique in that it has two world arguments, however if necessary, it could also be represented as a reified structure with each world argument as a separate property of that structure, where its truth value obtains in a single world.

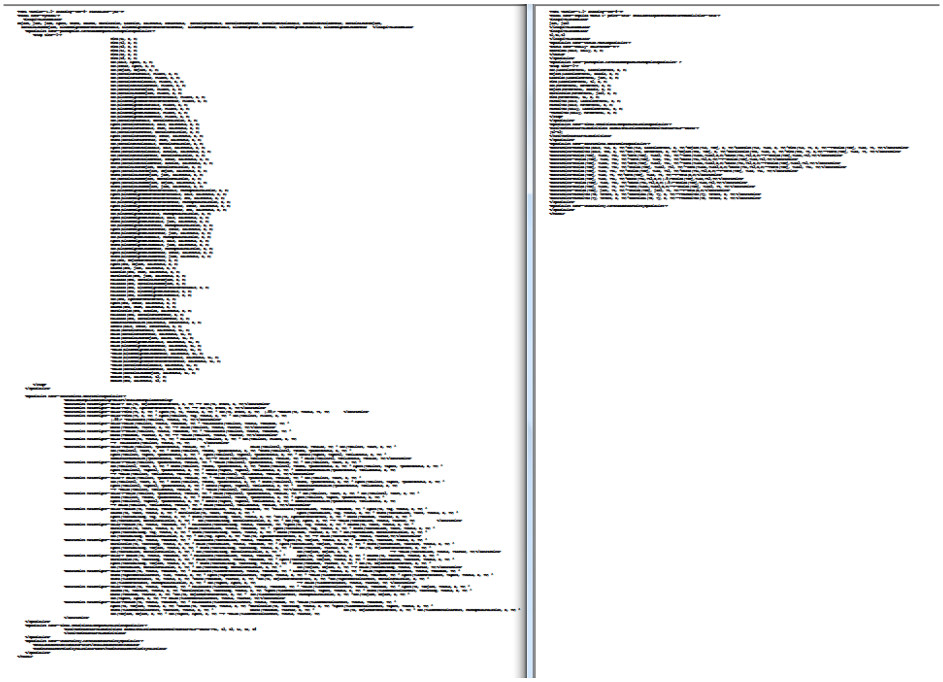

Figure 2: Polyscheme world reasoning models comparison

6.3 Significance of Results/Remaining Issues

Though many technical hurdles have been overcome, and though a working system is in place, the problem has not been fully solved. The first indicator is the fact that each term in a given constraint can generate multiple "implied matching" terms. Taken individually, these are not overwhelming, but taken together a relatively simple rule with 5 terms can explode to over 30. This adds an additional overhead to the matching process and in effect, creates an "upper bound" on the size and complexity of the reasoning that can be done. Another problem comes when we try to introduce implied matching over time through constraint conversion. None of the described approaches will handle the challenges which are raised by performing inference in a domain with multiple time steps. This is not merely a computational problem that must be solved over a set of discrete time units. Such a system needs to take into account all the ways in which times at different scales are related. This is complicated by the fact that human perceptions of the precision of time change as scales increase, making it hard to define boundaries over large intervals. (For example, what precisely is meant when I say that something happened "last year"?) In addition, properties which are said to hold at a given time step can "decay" in certainty as time goes on. If I am talking about the location of a mountain six months ago, I can be fairly confident that its position still holds. If I am talking about the freshness of fruit bought six months ago, I should have no such confidence. This example, in fact, points to a broader problem beyond that of integrating time. All of the examples described above had the shared quality of knowledge which was certain. The approach described has no ability to take a large set of uncertain terms, and from them extract the most likely or the "best" outcome. Instead, each factor must be considered independently in serial. The solution described above is a significant contribution to the existing Polyscheme architecture, but it also raises possibilities which point to a larger scale, and ultimately more sufficient solution.

7 Transitions

The project led to three transitions. Much of the technology developed here was key to the award of grants for the following three projects: 1. A MURI award for a project headed by Nicholas Cassimatis, the PI for the present project, to use these user modeling methods to improve human-computer interactions, 2. A DARPA SIBR with TracLabs that used these methods to monitor the operation of robots. 3. An ONR SIBR also with TracLabs that used these methods to greatly improve the ability of robots and understand the goals of the users they were interacting with.

The total amount of funding for these projects was about $8,000,000.

8 References

Barsalou, L. W. (2009). Simulation, situated conceptualization, and prediction. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364(1521), 1281-9.

Brooks, R. (1991). Intelligence without representation. *Artificial Intelligence*, 47, 139-159.

Baker, S. (February 20, 2011). 'Watson' has serious limitations. Retrieved from http://triblive.com/x/pittsburghtrib/opinion/columnists/guests/s_723648.html

Cassimatis, N. (2006). A cognitive substrate for achieving human-level intelligence. *AI Magazine*, 27(2), 45-56.

Cassimatis, N. L. (2006). Cognitive science and artificial intelligence have the same problem. Paper presented at the 2006 AAAI Spring Symposium: Between a rock and a hard place: Cognitive science principles meet AI-hard problems.

Cassimatis, N. (2008). Resolving Ambiguous, Implicit and Non-Literal References by Jointly Reasoning over Linguistic and Non-Linguistic Knowledge. *In Proceedings of SEMDIAL 2008.*

Cassimatis, N., Trafton, J. G., Bugajska, M., & Schultz, A. C. (2004). Integrating cognition, perception and action through mental simulation in robots. *Robotics and Autonomous Systems,* 49(1-2), 13-23.

Chalmers, D. J., French, R. M., & Hofstadter, D. R. (1992). High-level perception, representation, and analogy: A critique of artificial intelligence methodology. *Journal of Experimental & Theoretical Artificial Intelligence*, 4:185–211.

Cassimatis, N., Bignoli, P., Bugajska, M., Dugas, S., Kurup, U., Murugesan, A., et al. (2010). An architecture for adaptive algorithmic hybrids. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 1-13.

Crevier, D. (1993). *AI: The tumultuous history of the search for artificial intelligence*. New York: Basic Books.

Devlin, K. (1991). *Logic and information*. New York: Cambridge University Press.

Fauconnier, G. (1985). *Mental spaces*. Cambridge, MA: MIT Press.

Fauconnier, G. & M. Turner. (2002). *The way we think.* New York: Basic Books.

Goodman, N. (1947). The Problem of Counterfactual Conditionals. (F. Jackson, Ed.), 44(5), 113-128. Oxford University Press

Hayes, P. J. (1985). The Second naive physics mainfesto. In J. R. Hobbs & R. C. Moore, Formal Theories of the Commonsense World (pp. 1-36). Norwood, NJ: Ablex Publishing Company.

Lakoff, G., & Johnson, M. (1980). *Metaphors we live by*. Chicago: The University of Chicago Press.

Lakoff, G., & Johnson, M. (1999). *Philosophy in the flesh: The embodied mind and its challenge to western thought*. New York, NY: Basic Books. 10

Lewis, D. (1973). *Counterfactuals. Cambridge, MA: Harvard University Press.* McCarthy, J. (1996). From here to human-level AI. In *Proc. Of Principles of Knowledge Representation and Reasoning (KR).*

McShane, M., Nirenburg, S., and Beale, S. (2008). Two kinds of paraphrase in modeling embodied cognitive agents. In Proceedings of the Workshop on Biologically Inspired Cognitive Architectures, AAAI 2008 Fall Symposium, Washington, D.C., Nov. 7-9.

Scally, J.R., Cassimatis, N, and Uchida, H. (2012) Worlds as a Unifying Element of Knowledge Representation. *Biologically Inspired Cognitive Architectures*, 1, 14-22.

Stokes, Jon. (October 17, 2011) With Siri, Apple could eventually build a real AI. Retrieved from http://www.wired.com/insights/2011/10/with-siri-apple-could-eventually-build-a-real-ai.

Talmy, L. (2000). *Toward a cognitive semantics*. Cambridge, MA: MIT Press.